RESEARCH ARTICLE OPEN ACCESS

# Optimized Robot Path Planning Using Parallel Genetic Algorithm Based on Visible Midpoint

Motahar Reza, Saroj K.Satapathy, SubhashreePattnaik, Deepak Pandaand Prince Raj
*High Performance Computing Lab, School of Computer Science and Engineering, National Institute of Science and Technology, Berhampur,-761008, India*

**ABSTRACT**
An analysis is made for optimized path planning for mobile robot by using parallel genetic algorithm. The parallel genetic algorithm (PGA) is applied on the visible midpoint approach to find shortest path for mobile robot. The hybrid ofthese two algorithms provides a better optimized solution for smooth and shortest path for mobile robot. In this problem, the visible midpoint approach is used to make the effectiveness for avoiding local minima. It gives the optimum paths which are always consisting on free trajectories. But the proposedhybrid parallel genetic algorithm converges very fast to obtain the shortest route from source to destination due to the sharing of population. The total population is partitioned into a number subgroups to perform the parallel GA. The master thread is the center of information exchange and making selection with fitness evaluation.The cell to cell crossover makes the algorithm significantly good. The problem converges quickly with in a less number of iteration.
**Keywords:**GA,PGA,Robot pathplanning,cluttered environment

## I. INTRODUCTION

Path planning is a crucial problem in mobile robotics. A mobile robot must be able to searchcollision free paths to navigate from a start position to a goalposition in an environment with barriers.What path generated by robot, must be optimized under some certain criteria that is most important to the robot. The criteria may be shortest path, smoothest path or safest path.From the mid 80s, different predictable methods have been developed to solve the robotnavigation problem [1], such as global C-space methods [2], potential field methods [1], and neural networks approaches [3].Path planning problem can be solved by different conservative algorithms, such asthe potential field, Road Map cell decomposition [1].All most all approacheswere based on the concept ofconfiguration space. These approaches show lack of adaptation and a non robustactivities. Soeach method has its own strength over others in certain aspects, but also has some drawbacks. To get rid from the limitation of these procedures, researchers explored variety of solutions. In [4] ,Fuzzy Logic and Neural Network approach is implemented to handle the path planning problem. Genetic Algorithm (GA) is proposed in 1975 [5], andcan be employed to control a robot moving in a configuration space which has static obstacles and/or dynamic obstacle [6, 7].

GA has a greater impact on this field. The basic feature which makes GA's attractive in solving such types of problem, is that they are inherently parallel search techniques and can search for optimal in dynamic environments [7, 8] . GA can be used to optimize the path of a mobile robot moving in an environment which has number of static barriers. Some of the proposed methodologies in [9] suffer from different type of problems. They include (1) requires large memory spaces when dealing with dynamic andlarge sized environments (2) time consuming(3) computationally expensive . In the last decade, genetic algorithms have been widely used to generate the optimum path by intriguing the advantage of its strong optimization capability. This work is aggravated by our earlier work presented in [10].The parallel genetic Algorithm (PGA) gives good result and quickly converges to get the optimum result. There is number of processes available to implement PGA based on their information exchange [11]. A pseudo parallel GA can be used to find shortest path in a configuration space. The concept can be implemented by the number of chromosomes vary on the number of evolution [12]. In this study, we provide an initial idea based parallel genetic algorithm along with midpoint algorithm to select the shortest path in cluttered environment which will be able to handle static obstacles.

As per above discussion, in this paper, we propose a procedure to find an optimized path for robot in a cluttered environment by using PGA. This paper is organized as follows. Section 2 introduces problem statement for robot path planning (RPP). Section 3 discusses the proposed methodology for RPP. Section 4 presents the simulation results and demonstrates the solution for RPP. Finally, section 5 discusses some conclusions and considers possible future work.

## II. PROBLEM DESCRIPTION

The robot navigation planning problem is usually formulated as follows: a mobile robot is takenas a point robot and a complete description of configuration space which is 2-dimensional by nature. We must have to plan a obstacle free path between two specified locations, a source and a destination. The path should be a collision free and satisfies certain optimization criteria (i.e., shortest path). According to the above discussion, path planning problem is categorized as an optimization problem. Researchers uses various methods to solve the path planning problem as per the two common factors, the environment type (i.e., static or dynamic) and Path planning algorithms (i.e., global or local) . The Work spaceused for the problem is a static environment. The static environment is defined as the environment which doesn't contain any moving objects other than a navigating robot.Acomplete awareness about the search space and that all terrains of the space should be static, is required for implementing global path planning algorithms. Local path planning suggests that path planning is being implemented while the robot starts it's navigation; in other words, these algorithms provide the collision free movement in local region of robot.Fig.1 defines a model in which the black color objects are the obstacles in a 2-dimensional area with a start and goal point. The robot must move from start point to goal point without collision.

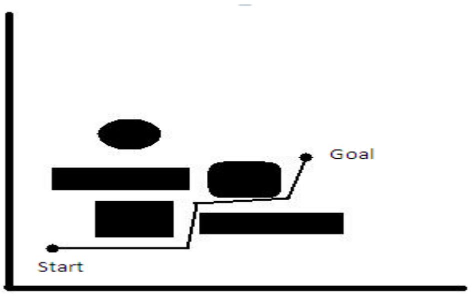


**Figure 1:**A Model for Robot path planning

## III. PROPOSED METHODOLOGY

In our proposed methodology, we have followed two steps to provide an optimized result. (1) Implement visible midpoint approach and (2) implement parallel genetic algorithm to optimize the path. The necessity of first step is to generate a graph where all paths available in configuration space lies on free trajectories available in configuration space. And in second step, the optimized path will be generating by the use of genetic algorithm.

### A. Visible midpoint approach

Visible midpoint approach is a type of classical path planning which provides a graph consists of all collision free paths in the configuration space. This approach provides two important activities:

1. This approach avoids the local minima in the configuration space.The Local minima is such a case where no results can be found in the configuration space.
2. The paths generated in the configuration space are the safest path because it doesn't touch the obstacle lines.

The cluttered terrains are the main reasons to put the robot in local minima situation. Local minima can be easily avoided by using visible midpoint algorithm because in each cluttered environment,there must be some paths which can helps robot to overcome the cluttered environment. Main focus of this algorithm is the generation of mid points in free trajectories. The mid points are generated according to the obstacles present in the environment. The paths are generated by connecting all the visible mid points available in configuration space and should not touch any obstacles present in the environment.Fig.2 represents the graph for line obstacle by using said algorithm and Fig.3 represents the graph for square type obstacles by the help of visible midpoint algorithm.

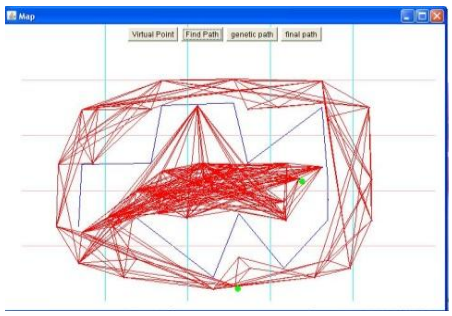


**Figure 2:**Visible midpoint output of Line Obstacle

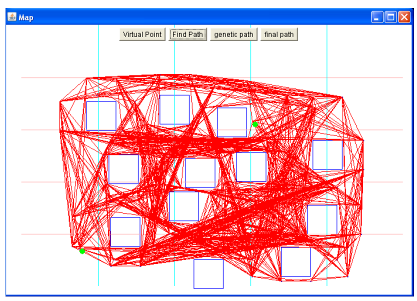


**Figure 3:**Visible midpoint output of square obstacle

**B. Optimized path by Parallel Genetic algorithm**

This step gives the optimum result for path planning. There are many parallelization-based methods for increasing the genetic algorithm execution speed. We have implemented master slave parallelism with genetic algorithm. The methodology takes theprimary population from the result provided by midpoint algorithm. The population contains number of individuals (i.e., chromosomes). Each candidate from the population represents a solution for the given problem under study. In our case, each solution is in fact a path from source to destination in the search space. The entire graph is decomposed into group of cells.. Each cells consists of a number of nodes and the information related to each cell is stored in a two dimensional matrix. The data for a single cell consists of the number of nodes is having along with the neighbor cells. Then the minimum edge connecting one cell to all other cells is stored in another adjacent matrix. Then the cell where the source and destination node is present is determined by traversing through the entire cell matrix. Then starting from the source node another node is determined by taking the minimum of all the nodes connected to it. When a node is selected for the path its duplicity is also checked with previously determined paths. If the node finds duplicity the node is discarded and other neighboring nodes are checked. Once the condition is satisfied the nodes is included in the path. The path is determined by connecting nodes to nodes and repeating the above procedure till the destination node is found. The entire process is repeated a number of times to get a desired number of paths.( 10 to 20 path) as initial population. Each path now represents a chromosome which is further optimized with the application of parallelgenetic algorithm.We have done the thread level parallelism to implement parallel GA. The master thread hold the responsibility of selection and fitness evaluation where as slaves are responsible for making crossover and mutation of genetic algorithm. In our proposed methodology, we have divided all the population into sub populations. Each sub population is assigned to each thread. After making crossover and mutation, the new population is submitted to master thread.

**1. Initialization**

The size of initial population is n and can be taken for the implementation purpose. Therefore the chromosomes are represented as {$C_1, C_2 \ldots . C_n$}.Each chromosome has its own fitness value and the fitness value is calculated as the sum of paths from start to goal. This means a path is the collection of paths from start to goal and can be represented as:

$P=\sum_{i=1}^{n} pi$ ……………………..(1)$pi=\sqrt{(x_{i+1}-x_i)^2-(y_{i+1}-y_i)^2}$……(2)

The eq.(1) represents the total length of path from a start point to goal point. Because a complete path is the summation of paths in the same trajectory from start point to goal point.Eq.(2) represents the length of a path segment from one point ($X_{i+1}$, $Y_{i+1}$) to another point ($X_i$, $Y_i$).

**2. Fitness function**

Fitness function performs a vital role in any evolutionary process using GA. Master Thread is having the responsibility to evaluate the fitness of each chromosome. Thequick search towards the optimal solution can be established by the appropriate selection of objective function. The optimal path is the shortest path fromsource to destination in our problem statement. Thus, the fitness function is having the responsibilityto locate the optimal path. The total number of intermediate path is may be the shortest path, which therobot needs to take to reach the destination. Therefore the cost of a path is the fitness function for our problem.Eq.(1) along with Eq.(2) can be used as the fitness function.

**3. Population Partition and Selection**

Whole population is equally divided into subpopulation (see Fig 4).Each set in the population is then submitted to an unique thread for further calculation. The subpopulation contains the chromosome pairs as per the selection made by the algorithm. Selection process is made according to the common points available in chromosome. If N is the total chromosome then

$N=\sum_{i=1}^{n} Ni$ ................…………………… (3)

Where Ni is the chromosome size. Each group contains even number of chromosomes.

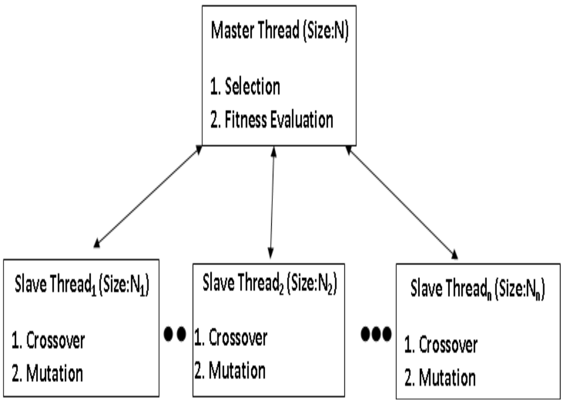


**Figure 4:** Create sub populations

**4. Crossover**

Then the crossover algorithm is applied to pair of chromosomes. The crossover is done by determining the crossover points for each pair. The crossover points are calculated by taking the cell common to both the chromosomes. The entire cell matrix is again traversed to find the cell common to both the chromosomes excluding the source cell and destination cell.

After determining the crossover point four nodes are determined as s1, d1, s2, d2 where s1 and d1 are the successor nodes and predecessor node respectively connected to the common cell's node of first chromosome and s2 and d2 are the successor nodes and predecessor node respectively connected to the common cell's node of second chromosome of a pair. The crossover node is determined to carry out the crossover procedure. The crossover node of the first child chromosome is determined by checking the existence of path from s1 as well as from d2 to the nodes of the common cell of second chromosome. If the cross over node is found then the first crossover path is calculated .The first crossover path is calculated by taking the source node till the successor node of first chromosome (s1) then crossover node and finally the predecessor node till the destination node of second chromosome (d2).

The crossover node of the second child chromosome is determined by checking the existence of path from s2 as well as from d2 to the node of the common cell of first chromosome. If the cross over node is found then the second crossover path is calculated .Then the second crossover is done by taking the source node till the successor node of second chromosome (s2) then crossover cell node and finally the predecessor node till the destination node of first chromosome (d1). Fig.5 represents the chromosome selected for crossover.Fig.6 represents the new chromosomes generated throw the help of crossover step.

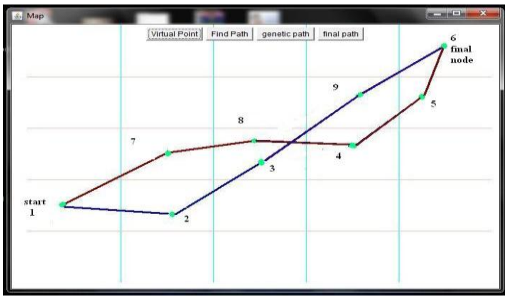


**Figure 5:**Crossover Step1 Output

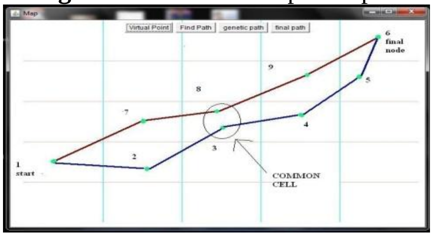


**Figure 6:**Crossover Step2 Output

**5. Mutation**

After getting the child chromosome pair the mutation operator is applied on the chromosomes. The nodes are interchanged in between the chromosomes to determine the new child chromosomes after mutation. Before the nodes are interchanged the existence of path for the new node is also checked.

Both the crossover and mutation method is applied to rest other pairs of chromosomes. After crossover of the entire parent set, all the chromosomes including the parent chromosomes and the new child set of chromosomes are again sorted according to the fitness function in ascending order. The new selected n-chromosomes are again submitted to master thread for further iteration.

## IV. ALGORITHM STEPS

\\M <-Cell Matrix (M1*M2)
\\N<-No. of chromosomes
\\$P_c$<-Crossover Probability
\\$P_m$<-Mutation Probability
ALGORITHM (M, N, Pc, Pm)

1. Start with population of n paths generated by Visible midpoint approach
2. Calculate the path distance from source to destination using eq.1 and eq.2 of each chromosome $N_i$ in the population.
3. Divide the population into subpopulation with equal size of chromosome. The subpopulations are submitted to sub threads for further evaluation.
4. Repeat the following steps until N offspring have been created in sub threads:

a. Select a pair of parent chromosomes from the current population.

b. With probability Pc crossover the pair at a randomly chosen point. If no crossover takes place, form two offspring that are exact copies of their respective parents. Crossover is done by cell wise manner. The crossover point is determined according to the common cell in cell matrix for both the chromosomes.
c. Mutate the two offspring at each locus with probability Pm and place the resulting chromosomes in the new population.

If N is odd, one new population member can be discarded at random.

5. Return the current population to Master thread.
6. If the solution is close to the optimum after several iteration, thengo to next step.
7. Else Go to Step-2.
8. End

## V. EXPERIMENTAL RESULT

We have implemented both the approach in a high performance computing Server with a Dell Power Edge R610Rack Server with Dual Intel XeonQuadcoreE5620 @2.93 GHz processors with a total 4 servers (64 Processors) 12GB RAM / 3x300 GB SAS HDD/ RAID – 5/RPS/ Dual NIC.TheMovementspace for robot is taken as 700*500 pixels. For our experiment, the graph is divided into group of 25 cells (5 * 5).Fig.7 represents the first step of proposed algorithm. In this experiment we have taken the obstacles as Square type.Fig.8 represents the second step that is optimized path from start point to goal point by using PGA. Fig.9 represents midpoint based visibility graph. In this experiment we have taken the obstacles as Line and which generates the Local Minima.Fig.10 represents the parallel genetic algorithm path from start point to goal point. In our experiment we have taken only 200 chromosomes which are then decomposed into 10 groups for 50th iterationswhere the chromosomes belong to different regions.Fig.11 shows the optimized value which is the distance from start to goal with respect to the number of iteration.GA provides the final result after $17^{th}$ iteration whereas PGA gives optimum result around $10^{th}$ iteration. The value is notgoing to change further.

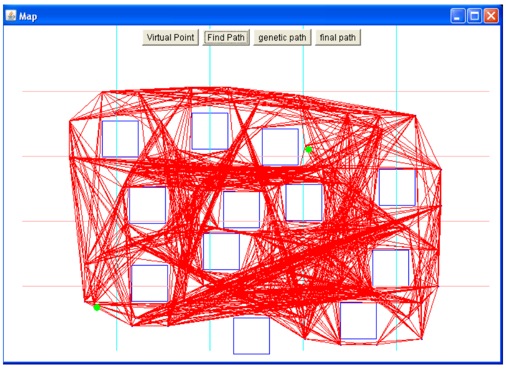


**Figure 7:** Visible midpoint output for Squared Obstacles

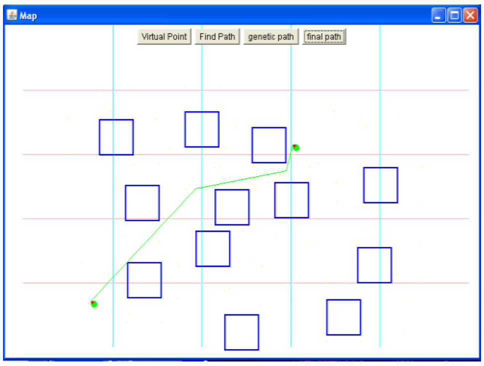


**Figure 8:** Optimized path By PGA for Squared Obstacles

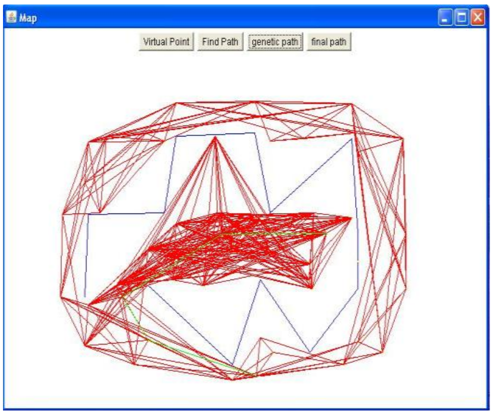


**Figure 9:**Visible midpoint output for Line Obstacles

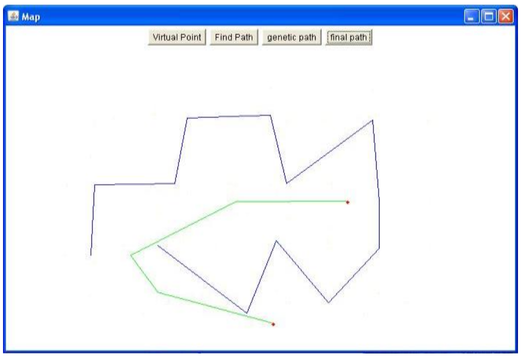


**Figure 10:**Optimized path By PGA for Line Obstacles

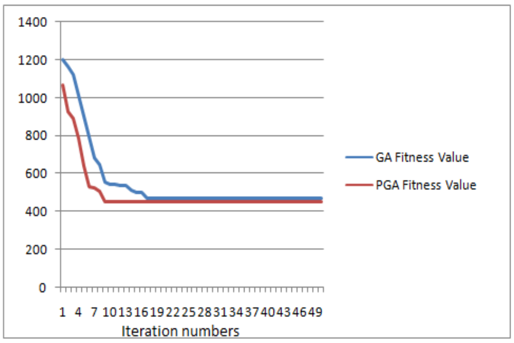


**Figure 11:**Fitness value vs Iteration

## VI. CONCLUSION

Parallel Genetic Algorithm is easy to apply on path planning problem because the path planning problem is a quite related to travelling Sales problem. The result, what we get from this implementation is effective as compared to classical path planning problem because the time complexity of classical path planning problem with Simple GA is high as compared to PGA. The problem converges quickly with in a less number of iteration. But we have seen that the mutation doesn't put any drastic changes on the result. The cell to cell Crossover makes the algorithm significantly good. The ultimate objective of developing a fully independent robot is beyond the scope of the research carried out for this dissertation. This paper necessarily focuses on a more specific component of an independent robot.

## REFERENCES

[1]. LatombeJ. C., Robot motion planning. Boston, Kluwer Academic Publisher, 1991.
[2]. SharirM., Algorithmic motion planning in robotics, Computer, vol. 22, no. 3, March 1989, 9-19.
[3]. ZhongY.Shirinzadeh, B. ; Y Tian.; A new neural network for robot path planning.Proceedings of Advanced Intelligent Mechatronics, 2008. AIM 2008. IEEE/ASME International Conference.Xian, July 2–5, pp. 1361–1366, 2008.
[4]. Huan Z, ZhiguoS ;Xiancui W. A rapid path planning adaptive optimization algorithm based on fuzzy neural network for multi-robot systems.IEEEconf.onCyberSpace Technology Beijing.Nov23,2013,pp 32-38
[5]. HollandJ., Adaptation in natural and artificial systems. Ann Arbor, University of Michigan Press, 1975.
[6]. Lee, J.; Kang, B.-Y. ; Kim, D.-W..; Fast genetic algorithm for robot path planning, published in Electronics Letters vol. 49,pp 1449-1451,2013.
[7]. Davis, L.; Handbook of Genetic Algorithms, VNR Comp. Library, 1990.
[8]. Dorigo, M. and Schnepf, U.; Genetic-based machine learning and behavior based robotics: A new synthesis, IEEE Transactions on Systems, Man, and Cybernetics, vol. 23, no. 1, pp. 141–154, 1993.
[9]. Ismail AL-Taharwa, AlaaSheta and Mohammed Al-Weshah, A Mobile Robot Path Planning UsingGenetic Algorithm in Static Environment. Journal of Computer Science, 4(4), 2008, 341-344.
[10]. Satapathy S.K.,M.Reza;A Visible Midpoint Approach for point Robot Path Planning in cluttered Environment NCCCS Nov21-22. 2012 Durgapur PP 1-4.
[11]. Luque, Gabriel, Alba, Enrique, Parallel Genetic Algorithms,Theory and Real World Applications Series: Studies in Computational Intelligence, Vol. 367,2011.
[12]. Cheng Shi ;Chen Shan-li,Path planning of robot soccer based improved pseudo-parallel genetic algorithm published in Optics Photonics and Energy Engineering (OPEE), 2010 IEEE conf.Vol-1 pp(243-246),Wuhan, China